AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application.

LISTING OF CLAIMS

Claims 1-180 (Cancelled).

181.   (New) A Hamming weight encoder, comprising:

an input that receives user data including P symbols;

a Hamming weight module that determines a Hamming weight of N of said P symbols, wherein N and P are integers greater than one and N is less than or equal to P;

a comparing module that compares the Hamming weight to a Hamming weight threshold; and

an inverting module that selectively bitwise inverts bits in said N symbols based on said comparison.

182.   (New) The Hamming weight encoder of Claim 181 wherein said Hamming weight encoder appends at least one inversion bit to said N symbols.

183.   (New) The Hamming weight encoder of Claim 181 wherein N and P are equal.

184. (New) A system comprising the Hamming weight encoder of Claim 181 and further comprising an error correction coding (ECC) module that performs ECC on at least one of said P symbols.

185. (New) The system of Claim 184 further comprising a run length limit (RLL) encoder that performs RLL encoding on an output of said ECC module.

186. (New) The system of Claim 185 wherein P is equal to N plus M, wherein said RLL encoder RLL encodes M symbols and does not RLL encode said N symbols, where M is an integer greater than zero.

187. (New) The system of Claim 186 wherein said Hamming weight module generates said Hamming weight for said N symbols and said comparing module compares said Hamming weight for said N symbols to said Hamming threshold.

188. (New) The Hamming weight encoder of Claim 181 wherein said Hamming weight is equal to a number of ones in said N symbols.

189. (New) The Hamming weight encoder of Claim 181 wherein said inverting module selectively inverts said N symbols when inverting increases said Hamming weight.

190. (New) The Hamming weight encoder of Claim 181 wherein said N symbols of said user data have a first Hamming weight, and wherein said N symbols output by said inverting module have a second Hamming weight that is greater than or equal to said first Hamming weight.

191. (New) A communications channel comprising:

a Hamming weight encoder comprising:

an input that receives user data including P symbols;

a Hamming weight module that determines a Hamming weight of N of said P symbols, wherein N and P are integers greater than one, P is equal to N plus M, and M is an integer greater than zero;

a comparing module that compares the Hamming weight to a Hamming weight threshold; and

an inverting module that selectively bitwise inverts bits in said N symbols based on said comparison; and

a run length limit (RLL) encoder that RLL encodes said M symbols and does not RLL encode said N symbols.

192. (New) The Hamming weight encoder of Claim 191 wherein said Hamming weight encoder appends at least one inversion bit to said P symbols.

193. (New) A system comprising the Hamming weight encoder of Claim 191 and further comprising an error correction coding (ECC) module that performs ECC on at least one of said P symbols.

194. (New) The Hamming weight encoder of Claim 191 wherein said Hamming weight is equal to a number of ones in said N symbols.

195. (New) The Hamming weight encoder of Claim 191 wherein said inverting module selectively inverts said N symbols when said inverting increases said Hamming weight.

196. (New) The Hamming weight encoder of Claim 191 wherein said N symbols of said user data have a first Hamming weight, and wherein said N symbols output by said inverting module have a second Hamming weight that is greater than or equal to said first Hamming weight.

197. (New) A Hamming weight encoder, comprising:

input means for receiving user data including P symbols;

Hamming weight means for determining a Hamming weight of N of said P symbols, wherein N and P are integers greater than one and N is less than or equal to P;

comparing means for comparing the Hamming weight to a Hamming weight threshold; and

inverting means for selectively bitwise inverting bits in said N symbols based on said comparison.

198.   (New)   The Hamming weight encoder of Claim 197 wherein said inverting means appends at least one inversion bit to said N symbols.

199.   (New)   The Hamming weight encoder of Claim 197 wherein N and P are equal.

200.   (New)   A system comprising the Hamming weight encoder of Claim 197 and further comprising error correction coding (ECC) means for performing ECC on at least one of said P symbols.

201.   (New)   The system of Claim 200 further comprising run length limit (RLL) encoding means for performing RLL encoding on an output of said ECC means.

202.   (New)   The system of Claim 201 wherein P is equal to N plus M, wherein said RLL encoding means RLL encodes said M symbols and does not RLL encode said N symbols, where M is an integer greater than zero.

203.   (New)   The system of Claim 202 wherein said Hamming weight means generates said Hamming weight for said N symbols and said comparing means compares said Hamming weight for said N symbols to said Hamming threshold.

204.   (New) The Hamming weight encoder of Claim 197 wherein said Hamming weight is equal to a number of ones in said N symbols.

205.   (New) The Hamming weight encoder of Claim 197 wherein said inverting means selectively inverts said N symbols when inverting increases said Hamming weight.

206.   (New) The Hamming weight encoder of Claim 197 wherein said N symbols of said user data have a first Hamming weight, and wherein said N symbols output by said inverting means have a second Hamming weight that is greater than or equal to said first Hamming weight.

207.   (New) A communications channel comprising:

Hamming weight encoding means for encoding comprising:

input means for receiving user data including P symbols;

Hamming weight means for determining a Hamming weight of N of said P symbols, wherein N and P are integers greater than one, M is an integer greater than zero, and P is equal to N plus M;

comparing means for comparing the Hamming weight to a Hamming weight threshold; and

inverting means for selectively bitwise inverting bits in said N symbols based on said comparison; and

run length limit (RLL) encoding means for RLL encoding said M symbols and not RLL encoding said N symbols.

208.   (New) The Hamming weight encoder of Claim 207 wherein said inverting means appends at least one inversion bit to said P symbols.

209.   (New) A system comprising the Hamming weight encoder of Claim 207 and further comprising error correction coding (ECC) means for performing ECC on at least one of said P symbols.

210.   (New) The Hamming weight encoder of Claim 207 wherein said Hamming weight is equal to a number of ones in said N symbols.

211.   (New) The Hamming weight encoder of Claim 207 wherein said inverting means selectively inverts said N symbols when said inverting increases said Hamming weight.

212.   (New) The Hamming weight encoder of Claim 207 wherein said N symbols of said user data have a first Hamming weight, and wherein said N symbols output by said inverting means have a second Hamming weight that is greater than or equal to said first Hamming weight.

213.   (New) A method for operating a Hamming weight encoder, comprising:

receiving user data including P symbols;

determining a Hamming weight of N of said P symbols, wherein N and P are integers greater than one and N is less than or equal to P;

comparing the Hamming weight to a Hamming weight threshold; and

selectively bitwise inverting bits in said N symbols based on said comparison.

214.　(New) The method of Claim 213 wherein said Hamming weight encoder appends inversion data to said N symbols.

215.　(New)　The method of Claim 213 wherein N and P are equal.

216.　(New) The method of Claim 213 further comprising performing ECC on at least one of said P symbols.

217.　(New)　The method of Claim 216 further comprising performing RLL encoding.

218.　(New)　The method of Claim 217 wherein P is equal to N plus M and further comprising RLL encoding said M symbols and not RLL encoding said N symbols, where M is an integer greater than zero.

219.　(New) The method of Claim 218 further comprising:

generating said Hamming weight for said N symbols; and

comparing said Hamming weight for said N symbols to said Hamming threshold.

220.   (New) The method of Claim 213 wherein said Hamming weight is equal to a number of ones in said N symbols.

221.   (New) The method of Claim 213 further comprising selectively inverting said N symbols when said inverting increases said Hamming weight.

222.   (New) The method of Claim 213 wherein said N symbols of said user data have a first Hamming weight, and wherein said N symbols after said inverting have a second Hamming weight that is greater than or equal to said first Hamming weight.

223.   (New) A communications channel comprising:

receiving user data including P symbols;

determining a Hamming weight of N of said P symbols, wherein N and P are integers greater than one, M is an integer greater than zero, and P is equal to N plus M;

comparing the Hamming weight to a Hamming weight threshold;

selectively bitwise inverting bits in said N symbols based on said comparison;

RLL encoding said M symbols; and

not RLL encoding said N symbols.

224.   (New)   The method of Claim 223 further comprising appending at least one inversion bit to said P symbols.

225.   (New)   The method of Claim 223 further comprising performing ECC on at least one of said P symbols.

226.   (New)   The Hamming weight encoder of Claim 223 wherein said Hamming weight is equal to a number of ones in said N symbols.

227.   (New)   The Hamming weight encoder of Claim 223 wherein said inverting selectively inverts said N symbols when said inverting increases said Hamming weight.

228.   (New)   The Hamming weight encoder of Claim 223 wherein said N symbols of said user data have a first Hamming weight, and wherein said N symbols output after said inverting have a second Hamming weight that is greater than or equal to said first Hamming weight.

229.   (New)   A method for operating a Hamming weight encoder, comprising:

receiving user data including P symbols;

determining a Hamming weight of N of said P symbols, wherein N and P are integers greater than one and N is less than or equal to P;

comparing the Hamming weight to a Hamming weight threshold; and

selectively bitwise inverting bits in said N symbols based on said comparison.

230.   (New) The method of Claim 229 wherein said Hamming weight encoder appends inversion data to said N symbols.

231.   (New) The method of Claim 229 wherein N and P are equal.

232.   (New) The method of Claim 229 further comprising performing ECC on at least one of said P symbols.

233.   (New)   The method of Claim 232 further comprising performing RLL encoding.

234.   (New) The method of Claim 233 wherein P symbols is equal to N plus M and further comprising RLL encoding said M symbols and not RLL encoding said N symbols, where M is an integer greater than zero.

235.   (New) The method of Claim 234 further comprising:

generating said Hamming weight for said N symbols; and

comparing said Hamming weight for said N symbols to said Hamming threshold.

236. (New) The method of Claim 229 wherein said Hamming weight is equal to a number of ones in said N symbols.

237. (New) The method of Claim 229 further comprising selectively inverting said N symbols when said inverting increases said Hamming weight.

238. (New) The method of Claim 229 wherein said N symbols of said user data have a first Hamming weight, and wherein said N symbols after said inverting have a second Hamming weight that is greater than or equal to said first Hamming weight.

239. (New) A method of operating a communications channel comprising:

receiving user data including P symbols;

determining a Hamming weight of N of said P symbols, wherein N and P are integers greater than one, M is an integer greater than zero, and P is equal to N plus M;

comparing the Hamming weight to a Hamming weight threshold;

selectively bitwise inverting bits in said N symbols based on said comparison;

RLL encoding said M symbols; and

not RLL encoding said N symbols.

240. (New) The method of Claim 239 further comprising appending at least one inversion bit to said P symbols.

241. (New) The method of Claim 239 further comprising performing ECC on at least one of said P symbols.

242. (New) The method of Claim 239 wherein said Hamming weight is equal to a number of ones in said N symbols.

243. (New) The method of Claim 239 wherein said inverting selectively inverts said N symbols when said inverting increases said Hamming weight.

244. (New) The method of Claim 239 wherein said N symbols of said user data have a first Hamming weight, and wherein said N symbols output after said inverting have a second Hamming weight that is greater than or equal to said first Hamming weight.

245. (New) A computer program stored for use by a processor for operating a Hamming weight encoder, comprising:

receiving user data including P symbols;

determining a Hamming weight of N of said P symbols, wherein N and P are integers greater than one and N is less than or equal to P;

comparing the Hamming weight to a Hamming weight threshold; and

selectively bitwise inverting bits in said N symbols based on said comparison.

246.   (New) The computer program of Claim 245 wherein said Hamming weight encoder appends inversion data to said N symbols.

247.   (New) The computer program of Claim 245 wherein N and P are equal.

248.   (New) The computer program of Claim 245 further comprising performing ECC on at least one of said P symbols.

249.   (New) The computer program of Claim 248 further comprising performing RLL encoding.

250.   (New) The computer program of Claim 249 wherein P is equal to N plus M and further comprising RLL encoding said M symbols and not RLL encoding said N symbols, where M is an integer greater than zero.

251.   (New) The computer program of Claim 250 further comprising:

generating said Hamming weight for said N symbols; and

comparing said Hamming weight for said N symbols to said Hamming threshold.

252.   (New) The computer program of Claim 245 wherein said Hamming weight is equal to a number of ones in said N symbols.

253.   (New) The computer program of Claim 245 further comprising inverting said N symbols when said inverting increases said Hamming weight.

254.   (New) The computer program of Claim 245 wherein said N symbols of said user data have a first Hamming weight, and wherein said N symbols after said inverting have a second Hamming weight that is greater than or equal to said first Hamming weight.

255.   (New) A computer program stored for use by a processor for operating a communications channel comprising:

receiving user data including P symbols;

determining a Hamming weight of N of said P symbols, wherein N and P are integers greater than one, M is an integer greater than zero, and P is equal to N plus M;

comparing the Hamming weight to a Hamming weight threshold;

selectively bitwise inverting bits in said N symbols based on said comparison;

RLL encoding said M symbols; and

not RLL encoding said N symbols.

256. (New) The computer program of Claim 255 further comprising appending at least one inversion bit to said P symbols.

257. (New) The computer program of Claim 255 further comprising performing ECC on at least one of said P symbols.

258. (New) The computer program of Claim 255 wherein said Hamming weight is equal to a number of ones in said N symbols.

259. (New) The computer program of Claim 255 wherein said inverting selectively inverts said N symbols when said inverting increases said Hamming weight.

260. (New) The computer program of Claim 255 wherein said N symbols of said user data have a first Hamming weight, and wherein said N symbols output after said inverting have a second Hamming weight that is greater than or equal to said first Hamming weight.